Simple Greedy Algorithms for Fundamental Multidimensional Graph Problems*

Vittorio Bilò[1], Ioannis Caragiannis[2], Angelo Fanelli[3], Michele Flammini[4], and Gianpiero Monaco[5]

1 **Department of Mathematics and Physics "Ennio De Giorgi", University of Salento, Salento, Italy**
vittorio.bilo@unisalento.it
2 **CTI "Diophantus" & Department of Computer Engineering and Informatics, University of Patras, Patras, Greece**
caragian@ceid.upatras.gr
3 **CNRS (UMR-6211), Caen, France**
angelo.fanelli@unicaen.fr
4 **Gran Sasso Science Institute & DISIM, University of L'Aquila, L'Aquila, Italy**
michele.flammini@univaq.it
5 **DISIM, University of L'Aquila, L'Aquila, Italy**
gianpiero.monaco@univaq.it

Abstract

We revisit fundamental problems in undirected and directed graphs, such as the problems of computing spanning trees, shortest paths, steiner trees, and spanning arborescences of minimum cost. We assume that there are d different cost functions associated with the edges of the input graph and seek for solutions to the resulting multidimensional graph problems so that the p-norm of the different costs of the solution is minimized. We present combinatorial algorithms that achieve very good approximations for this objective. The main advantage of our algorithms is their simplicity: they are as simple as classical combinatorial graph algorithms of Dijkstra and Kruskal, or the greedy algorithm for matroids.

1998 ACM Subject Classification F.2.2 [Nonnumerical Algorithms and Problems] Computations on Discrete Structures, G.2.2 [Graph Theory] Graph Algorithms

Keywords and phrases multidimensional graph problems, matroids, shortest paths, Steiner trees, arborescences

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.125

1 Introduction

We study generalizations of some very well-known combinatorial optimization problems, such as the problem of computing a minimum spanning tree in a graph. In its classical version, we are given an undirected graph with edge costs and the objective is to compute a spanning tree of minimum cost on the graph. We revisit fundamental problems of this kind by assuming that there are d different cost functions associated with the edges of the input graph. Then, a spanning tree has d different cost values, one for each cost function. Our objective is to compute a spanning tree that minimizes a specific aggregate value of these costs.

* This work was partially supported by the project ANR-14-CE24-0007-01 *"CoCoRICo-CoDec"*.

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 125; pp. 125:1–125:13

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

At first glance, this is the type of multi-objective (or multi-criteria) optimization problems that arise in many diverse disciplines, including engineering, economics and business, healthcare, and more. Here, our motivation stems from the recently emerging trend of *participatory budgeting* [1]. According to it, optimization problems related to the use of budget for building public facilities are solved taking into account the view the citizens have for the input. In the example above, each of the d different cost functions can be thought of as provided by a single individual. Consistent to this, we will assume that the parameter d is large.

We use the general term *minimum multidimensional resource selection* (MMRS) to refer to the class of problems that we study. In addition to parameter d, an instance of such a problem consists of a set R of resources, a d-dimensional cost vector $\boldsymbol{c}_r$ for each resource $r \in R$, the set of feasible solutions $\mathcal{F}$ (subsets of resources) and an additional parameter $p \geq 1$. The objective of MMRS is to select a feasible solution S so that the quantity $\| \sum_{r \in S} \boldsymbol{c}_r \|_p$ is minimized. Note that the sum inside the norm is d-dimensional and its entries represent the cost of S with respect to the d cost functions. Then, the p-norm is used for aggregating these entries into a single value.

Even though the problem has not been considered before in the general version we just defined it, efficient solutions to some of its variants follow by recent advances on randomized rounding of fractional solutions for linear program relaxations. In contrast to such sophisticated techniques, we insist on deterministic algorithms that are extremely simple. More concretely, we consider the following problems:

- We warm up with MMRS in *matroids*, in which the feasible solutions that form set $\mathcal{F}$ are the bases of a matroid defined over the resources. A typical subproblem is when the resources are the edges of a graph and the feasible solutions correspond to spanning trees of the graph. For MMRS on matroids, we present a variation of the greedy algorithm on matroids (e.g., see [18]) and show that it yields $\mathcal{O}(\min\{p, \log d\})$-approximate solutions.
- *Shortest multidimensional path* (SMP). Again, the resources correspond to edges of a graph and the feasible solutions are subsets of edges that connect two designated nodes. An approximation guarantee of $\mathcal{O}(\min\{p, \log d\})$ is obtained by a Dijkstra-like algorithm.
- *Minimum multidimensional steiner tree* (MMST). Unlike the spanning tree version mentioned above, in MMST the feasible solutions are not matroid bases. Furthermore, the classical trick in the single dimensional case (see, e.g., [19]) of approximating the minimum steiner tree by a minimum spanning tree does not carry over when we have different costs (the cost functions do not necessarily form a metric). Still, we have a Kruskal-like algorithm that uses our shortest multidimensional path algorithm as subroutine and achieves (asymptotically) the same approximation guarantee.
- *Minimum multidimensional arborescence* (MMA). Here, the resources are the edges of a directed graph and the feasible solutions are spanning trees, directed away from a designated root node. We present another simple algorithm that uses our shortest multidimensional path algorithm as a subroutine and prove it to be $\mathcal{O}(\min\{p, \log d\} \cdot \log n)$-approximate, where n is the graph size.

We complement these results with an inapproximability statement. For $p = \infty$, none of the above problems admit a polynomial-time constant approximation algorithm, under standard complexity assumptions. Here we exploit a gap-preserving reduction from the vector scheduling problem which has been proved to be inpproximable in [7].

Our analysis is inspired by the literature on online scheduling and in particular from [2, 5] where the objective is to minimize the p-norm of machine loads. En route, we exploit a nice structural property that is satisfied by feasible solutions of the problems that we study, and implies that greedy solutions for them are efficient.

Related work. The field of multi-objective optimization has traditionally considered similar problems to ours, with approximation algorithms playing a major role. It seems though that the questions considered there are mostly related to approximating the Pareto-curve (i.e., the set of solutions which are not dominated by any other). For instance, Diakonikolas et al. [11] (see also the references therein) study the problem of computing a minimum set of solutions that approximates within a specified accuracy, $\epsilon > 0$, the Pareto curve of bi-objective optimization problems, containing many important widely studied problems such as shortest paths, spanning tree, matching, etc. We notice that these questions are fundamentally different than the ones we study here. Also, they turn out to be computationally meaningful only on instances in which d is a small constant (since otherwise the problem becomes notoriously hard due to fact that the Pareto-curve becomes huge [16]). Instead, we are interested in many different cost functions.

Chekuri et al. [8] (see also [9]) study the problem of solving (or, better, approximating the optimal solution of) *minimax integer programs* subject to a matroid constraint; this is essentially what we call MMRS on matroids with a value of infinity for parameter p. Chekuri et al. [8] present a $\mathcal{O}(\log d/\log\log d)$-approximation algorithm for this problem that exploits sophisticated randomized rounding techniques. We remark that our bound is slightly higher than theirs but the advantage of our result is in the simplicity of the algorithm. A special case is covered in [4], where the computation of a spanning tree minimizing the maximum number of times its edges cross a given set of cuts is considered.

Other investigations related to our setting are the multi-budgeted optimization problems. There are d different cost functions defined over the set of resources. In addition, there are d budget values that constrain each of the d costs of a solution. The objective is to compute a feasible solution whose budget violation factor is as small as possible across all cost dimensions. These problems have been tackled using sophisticated approaches such as Lagrangian relaxations combined with various technical properties of the underlying combinatorial structure, and linear programming together with iterative rounding techniques. Such approaches were used to develop PTASes for spanning trees [14, 17], shortest paths [13, 15], and matchings and matroid intersection [3] with $d = 2$. Grandoni et al. [12] consider d-budgeted versions of classical problems. They show PTASes for spanning trees, matroid bases, and bipartite matchings. Moreover they get a deterministic approximation scheme for d-budgeted matchings in general graphs. We emphasize that the authors of [12] use linear programming formulations and iterative rounding techniques that work for constant values of d only. Finally, Chekuri et al. [10] give a randomized PTAS for matroid intersection and matchings with any fixed number of budget constraints.

Roadmap. We begin with some necessary mathematical background in Section 2. Our main lemma is presented in Section 3. Then, Sections 4-7 are devoted to the each of the four problems mentioned above. We conclude with our inapproximability result in Section 8.

2 Mathematical Background

We summarize definitions and simple properties of p-norms and matroids. For an integer $d \geq 1$, define $[d] = \{1, 2, \ldots, d\}$ and $\mathbf{0}^d$ as the vector $(0, \ldots, 0)^T \in \mathbb{R}^d$. We denote by $\overline{\mathbb{R}} = \mathbb{R}_{\geq 1} \cup \{\infty\}$ the set of reals that are higher than 1, extended with the value ∞.

We later exploit the following two properties of the function $f(x) = x^t$ for every $t \geq 1$.

▶ **Lemma 1.** *For every $x, y, h \geq 0$ with $y \geq x$ and $t \geq 1$, $(x + h)^t - x^t \leq (y + h)^t - y^t$.*

▶ **Lemma 2.** *Given $x \geq 0$, $t \geq 1$, and n non-negative real values $h_1, \ldots, h_n$,*

$$\sum_{i\in[n]} \left((x+h_i)^t - x^t\right) \leq \left(x + \sum_{i\in[n]} h_i\right)^t - x^t.$$

The first property comes by observing that $(x+h)^t - x^t$ has non-negative derivative with respect to x when $t \geq 1$, while the second one is due to convexity of the monomial x^t (see [6] for a proof).

For a vector $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and $p \in \overline{\mathbb{R}}$, the value $\|\boldsymbol{x}\|_p = \left(\sum_{i\in[d]} x_i^p\right)^{1/p}$ is called the *p-norm* of $\boldsymbol{x}$. We recall two fundamental properties possessed by p-norms.

▶ Property 3. For every $\boldsymbol{x} \in \mathbb{R}^d$ and $p, p' \in \overline{\mathbb{R}}$ such that $p' \leq p$, $\|\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_{p'}$.

▶ Property 4 (Minkowsi's Inequality). For every $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $p \in \overline{\mathbb{R}}$, $\|\boldsymbol{x}+\boldsymbol{y}\|_p \leq \|\boldsymbol{x}\|_p + \|\boldsymbol{y}\|_p$.

The next lemma shows how to use the logarithmic norm to approximate all other p-norms; its proof follows easily by the definitions.

▶ **Lemma 5.** *For every $\boldsymbol{x} \in \mathbb{R}^d$ and $p \in \overline{\mathbb{R}}$, $\|\boldsymbol{x}\|_{\ln d} \leq e\|\boldsymbol{x}\|_p$.*

A *matroid* is a pair $M = (R, \mathcal{X})$ such that R is a finite set, called the *ground set*, and $\mathcal{X}$ is a family of subsets of R with the following properties:

1. $\emptyset \in \mathcal{X}$,
2. if $X \in \mathcal{X}$ and $Y \subset X$, then $Y \in \mathcal{X}$ (*hereditary property*),
3. if $X, Y \in \mathcal{X}$ and $|X| > |Y|$, then there exists $x \in X \setminus Y$ such that $Y \cup \{x\} \in \mathcal{X}$ (*independent set exchange property*).

A *basis* for matroid M is a set $B \in \mathcal{X}$ such that $B \cup \{x\} \notin \mathcal{X}$ for every $x \in R \setminus B$. The independent set exchange property implies that all bases of M have the same cardinality which is called the *rank* of M and is denoted by $r(M)$.

For every two bases $B_1, B_2 \in \mathcal{X}$, denote by $G(B_1 \Delta B_2)$ the bipartite graph (V, E) such that $V = (B_1 \setminus B_2) \cup (B_2 \setminus B_1)$ and $E = \{\{e_1, e_2\} : e_1 \in B_1 \setminus B_2, e_2 \in B_2 \setminus B_1, B_1 \setminus \{e_1\} \cup \{e_2\} \in \mathcal{X}\}$. We shall make extensive use of the following fundamental result (see [18]).

▶ Proposition 6. There exists a perfect matching in the graph $G(B_1 \Delta B_2)$.

Given an undirected graph $G = (V, E)$, let $\mathcal{X}$ be the family of all subsets of E which do not contain cycles. The pair $M = (E, \mathcal{X})$ is a matroid and is called the *graphic matroid* defined over G. The set of bases for M is the set of all spanning trees for G, so that $r(M) = |V| - 1$.

3 Problem Statement and the PAID Property

The minimum multidimensional resource selection (MMRS) problem is a collection of instances of the form $I = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}, p)$, where R is a set of resources such that each resource $r \in R$ has an associated d-dimensional cost vector $\boldsymbol{c}_r \in \mathbb{R}^d_+$, $\mathcal{F} \subseteq 2^R \setminus \emptyset$ is a set of feasible solutions, and $p \in \overline{\mathbb{R}}$. For a subset of resources $S \subseteq R$, define its multidimensional load as $\boldsymbol{\ell}(S) = \sum_{r\in S} \boldsymbol{c}_r$ and denote by $\ell_i(S)$ its ith element. An optimal solution for I is any solution belonging to $\operatorname{argmin}_{S\in\mathcal{F}} \left\{\|\boldsymbol{\ell}(S)\|_p\right\}$, that is, any feasible solution minimizing the p-norm of its multidimensional load. Denote by $\mathsf{OPT}(I) = \|\boldsymbol{\ell}(S^*)\|_p$, where $S^* \in \operatorname{argmin}_{S\in\mathcal{F}} \left\{\|\boldsymbol{\ell}(S)\|_p\right\}$, the p-norm of the multidimensional load of an optimal solution for I.

We shall denote by $\mathsf{MMRS}(p)$ the natural restriction of the MMRS obtained by fixing the value of p. When referring to an instance $I \in \mathsf{MMRS}(p)$, we remove the value of p from the description of I and simply write $I = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F})$.

Given a set X and an integer $k \geq 1$, a *partial k-decomposition* of X is an ordered family of k sets $(X_1, \ldots, X_k)$ such that $\bigcup_{i\in[k]} X_i \subseteq X$. A *$\beta$-intersecting partial k-decomposition* of X is a partial k-decomposition of X $(X_1, \ldots, X_k)$ such that $|\{i \in [k] : x \in X_i\}| \leq \beta$ for every $x \in X$, that is, no element of X occurs in more than β components of the decomposition. A k-decomposition of X is a partial *k-decomposition* of X $(X_1, \ldots, X_k)$ such that $\bigcup_{i\in[k]} X_i = X$. A *$k$-partition* of X is a 1-intersecting k-decomposition of X.

▶ **Definition 7.** Fix an instance $I \in \mathsf{MMRS}(p)$. A feasible solution S for I has the *pairwise β-intersecting decomposition* property (henceforth, β-PAID property) if there exist a k-decomposition of S $(S_1, \ldots, S_k)$ and a β-intersecting partial k-decomposition of an optimal solution S^* for I $(S_1^*, \ldots, S_k^*)$ such that, for every $i \in [k]$,

$$\|\boldsymbol{\ell}(S_{\leq i})\|_p \leq \|\boldsymbol{\ell}(S_{\leq i-1}) + \boldsymbol{\ell}(S_i^*)\|_p\,, \tag{1}$$

where $S_{\leq i} = \bigcup_{j\in[i]} S_j$ and $S_{\leq 0} = \emptyset$.

The importance of the PAID property is captured by the following result.

▶ **Lemma 8.** *Fix an instance $I \in \mathsf{MMRS}(p)$. If a feasible solution S for I possesses the β-PAID property, then $\|\boldsymbol{\ell}(S)\|_p \leq \frac{\beta p}{\ln 2}\mathsf{OPT}(I)$.*

Proof. We get

$$\begin{aligned}
\left(\|\boldsymbol{\ell}(S)\|_p\right)^p &= \sum_{j\in[d]} \ell_j(S)^p = \sum_{i\in[k]} \left(\sum_{j\in[d]} \ell_j(S_{\leq i})^p - \sum_{j\in[d]} \ell_j(S_{\leq i-1})^p\right)\\
&\leq \sum_{i\in[k]} \left(\sum_{j\in[d]} \left(\ell_j(S_{\leq i-1}) + \ell_j(S_i^*)\right)^p - \sum_{j\in[d]} \ell_j(S_{\leq i-1})^p\right)\\
&= \sum_{j\in[d]}\sum_{i\in[k]} \left(\left(\ell_j(S_{\leq i-1}) + \ell_j(S_i^*)\right)^p - \ell_j(S_{\leq i-1})^p\right)\\
&\leq \sum_{j\in[d]}\sum_{i\in[k]} \left(\left(\ell_j(S) + \ell_j(S_i^*)\right)^p - \ell_j(S)^p\right)\\
&\leq \sum_{j\in[d]} \left(\left(\ell_j(S) + \sum_{i\in[k]} \ell_j(S_i^*)\right)^p - \ell_j(S)^p\right)\\
&\leq \sum_{j\in[d]} \left(\left(\ell_j(S) + \beta\ell_j(S^*)\right)^p - \ell_j(S)^p\right)\\
&= \sum_{j\in[d]} \left(\ell_j(S) + \beta\ell_j(S^*)\right)^p - \sum_{j\in[d]} \ell_j(S)^p\\
&\leq \left(\|\boldsymbol{\ell}(S)\|_p + \beta\|\boldsymbol{\ell}(S^*)\|_p\right)^p - \left(\|\boldsymbol{\ell}(S)\|_p\right)^p.
\end{aligned}$$

The first inequality follows by raising both sides of inequality (1) to p. The second and third inequalities follow from Lemmas 1 and 2, respectively. The fourth inequality holds since $(S_1^*, \ldots, S_k^*)$ is a β-intersecting partial k-decomposition of S^*. The fifth inequality follows by Minkowski's inequality (by raising both of its sides to p).

By rearranging, we obtain $\left(2^{1/p} - 1\right)\|\boldsymbol{\ell}(S)\|_p \leq \beta\|\boldsymbol{\ell}(S^*)\|_p$, which implies

$$\|\boldsymbol{\ell}(S)\|_p \leq \frac{\beta\|\boldsymbol{\ell}(S^*)\|_p}{2^{1/p} - 1} = \frac{\beta\|\boldsymbol{\ell}(S^*)\|_p}{e^{\ln 2/p} - 1} \leq \frac{\beta p}{\ln 2}\|\boldsymbol{\ell}(S^*)\|_p,$$

Algorithm 1 $(M, d, (\boldsymbol{c}_r)_{r\in R}, p)$

$S_0 \leftarrow \emptyset$
$i \leftarrow 0$
while $S_{\leq i}$ is not a basis for M **do**
  $i \leftarrow i + 1$
  $C_i \leftarrow \{x \in R : S_{\leq i-1} \cup \{x\} \in \mathcal{X}\}$
  $S_i \leftarrow \operatorname{argmin}_{x\in C_i} \|\boldsymbol{\ell}(S_{\leq i-1} \cup \{x\})\|_p$
end while
$S \leftarrow S_{\leq i}$
return S

where the last inequality comes from the fact that $e^x \geq x + 1$ for every $x \geq 0$. Since S^* is an optimal solution for I, the claim follows. ◀

By exploiting Lemma 5, we get the following approximability result; the proof is omitted due to lack of space.

▶ **Lemma 9.** *Let S be an α-approximate solution to an instance $I = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}) \in$ MMRS$(\ln d)$. Then, S is an $O(\alpha)$-approximate solution to the instance $\overline{I} = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}) \in$ MMRS(p) with $p \geq \ln d$.*

By putting all together, we get the following general approximation theorem.

▶ **Theorem 10.** *Let A be an algorithm which, for every instance $I \in$ MMRS, computes a feasible solution for I possessing the β-PAID property. Then, A approximates* MMRS *within a factor of $O(\beta \cdot \min\{p, \log d\})$.*

Proof. Fix an instance $I = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}, p) \in$ MMRS. We can use algorithm A to obtain a feasible solution S for I possessing the β-PAID property. By Lemma 8, S is an $O(p\beta)$-approximate solution for I. Moreover, we can use algorithm A to obtain a feasible solution S for the instance $I' = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}, \ln d)$ possessing the β-PAID property. By Lemma 8, S is an $O(\beta \log d)$-approximate solution for I' so that, by Lemma 9, S is also an $O(\beta \log d)$-approximate solution for I. ◀

4 MMRS on Matroids

As a warmup application of our technique, we first consider instances $(R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}, p) \in$ MMRS such that $\mathcal{F}$ is the set of bases of a matroid $M = (R, \mathcal{X})$. We propose a simple greedy algorithm (Algorithm 1) to approximate MMRS in this case.

The following lemma characterizes the approximation guarantee achieved by Algorithm 1.

▶ **Lemma 11.** *Fix an instance $I = (R, d, (\boldsymbol{c}_r)_{r\in R}, \mathcal{F}, p) \in$ MMRS such that $\mathcal{F}$ is the set of bases of a matroid $M = (R, \mathcal{X})$. Algorithm 1 returns a feasible solution for I possessing the* 1-PAID *property.*

Proof. The fact that Algorithm 1 terminates by returning a feasible solution $S \in \mathcal{F}$ (i.e., a basis for M) follows from the classical analysis of the greedy algorithm for matroids. Set $k = r(M)$ and let S^* be an optimal solution to I. Define $R(S, S^*) = S \cap S^*$ and consider graph $G(S \Delta S^*)$. By Proposition 6, there exists a bijective function $f : S \setminus R(S, S^*) \to S^* \setminus R(S, S^*)$.

Note that Algorithm 1 implicitly defines a k-partition $(S_1, \ldots, S_k)$ of S. Now define the k-partition $(S_1^*, \ldots, S_k^*)$ of S^* such that, for every $i \in [k]$,

$$S_i^* = \begin{cases} S_i & \text{if } S_i \in R(S, S^*), \\ f(S_i) & \text{if } S_i \notin R(S, S^*). \end{cases}$$

Fix an index $i \in [k]$. We claim that, at the ith iteration of the while-loop of Algorithm 1, $S_{\leq i-1} \cup \{S_i^*\} \in C_i$. In fact, if this is not the case, it must be $S_{\leq i-1} \cup \{S_i^*\} \notin \mathcal{X}$. However, by the definition of f, we have that $S \setminus \{S_i\} \cup \{S_i^*\} \in \mathcal{X}$ which, by the hereditary property of matroids, implies that $S_{\leq i-1} \cup \{S_i^*\} \in \mathcal{X}$: a contradiction. Now, given that $S_i^* \in C_i$, the greedy choice performed at line 6 of Algorithm 1 implies that $\|\boldsymbol{\ell}(S_{\leq i})\|_p \leq \|\boldsymbol{\ell}(S_{\leq i-1} \cup S_i^*)\|_p \leq \|\boldsymbol{\ell}(S_{\leq i-1}) + \boldsymbol{\ell}(S_i^*)\|_p$. ◀

By combining the above lemma with Theorem 10, we obtain the following result.

▶ **Theorem 12.** *Algorithm 1 approximates* MMRS *on matroids within a factor of* $O(\min\{p, \log d\})$.

5 Shortest Multidimensional Path

Given a directed graph $G = (V, E)$, in which every edge $e \in E$ is associated with a d-dimensional weight $\boldsymbol{c}_e \in \mathbb{R}_+^d$, a pair $(s, t) \in V^2$ of source-destination nodes, and value $p \in \overline{\mathbb{R}}$, the shortest multidimensional path (SMP) problem is the restriction of MMRS to instances with $R = E$ and $\mathcal{F} = \{S \subseteq E : S \text{ is an } (s,t)\text{-path in } G\}$.

For our purposes, we shall need to solve instances of the SMP when dealing with other MMRSs on graphs. For such a reason, we shall define an approximation algorithm for the SMP which requires more general input parameters than the ones needed to solve the SMP.

Towards this end, consider the multidimensional generalization of Dijkstra's algorithm, denoted as Algorithm 2, defined in the following. It takes as input the graph G, the integer d, the pair of nodes (s, t), the value p, and a set of edges E' which may contain either edges in E and edges not in E, and makes use of the data structures PATH and DISTANCE. Given a node $v \in V$, PATH is an array such that PATH$[v]$ contains a path connecting s to v, and DISTANCE is an array such that DISTANCE$[v]$ contains the value $\|\boldsymbol{\ell}(\text{PATH}[v] \cup E')\|_p$.

The following lemma characterizes the approximation guarantee achieved by Algorithm 2.

▶ **Lemma 13.** *Fix an instance $I = (G, d, s, t, p) \in$ SMP. Then, Algorithm 2, executed with parameters G, d, s, t, p and $\emptyset$, returns a feasible solution for I possessing the* 1-PAID *property.*

Proof. The fact that, when executed with parameters G, d, s, t, p and $\emptyset$, Algorithm 2 terminates by returning a set of edges $S = \text{PATH}[t]$ inducing an (s,t)-path in G follows from the classical analysis of Dijkstra's algorithm. Hence, we only need to show that S possesses the 1-PAID property.

Let S^* be an optimal solution to I. For a node v and a path P, let $pred_P(v)$ be the predecessor of node v along P. A node v is a *merging node* for S and S^* if (i) $v \in \{s, t\}$ or (ii) v occurs along both S and S^* and $pred_S(v) \neq pred_{S^*}(v)$. Denote by $M(S, S^*) = (s = v_0, v_1, \ldots, v_j = t)$ the sequence of merging nodes for S and S^* numbered according to the order in which they occur along S. A node $v_i \in M(S, S^*)$ is *redundant* if v_i occurs along S^* before some other merging node v_j with $j < i$. Denote by $\overline{M}(S, S^*) = (s = \overline{v}_0, \overline{v}_1, \ldots, \overline{v}_k = t)$ the sequence of nodes obtained from $M(S, S^*)$ by removing all the redundant ones (see Figure 1 for an illustrating example).

Algorithm 2 (G, d, s, t, p, E')

```
1: for each v ∈ V do
2:     PATH[v] ← ∅
3:     DISTANCE[v] ← +∞
4: end for
5: DISTANCE[s] ← ‖ℓ(E′)‖_p
6: Q ← V
7: while Q is not empty do
8:     v ← argmin_{u∈Q}{DISTANCE[u]}
9:     for each (v, u) ∈ E do
10:        if DISTANCE[u] > ‖ℓ(PATH[v] ∪ (v, u) ∪ E′)‖_p then
11:            PATH[u] ← PATH[v] ∪ (v, u)
12:            DISTANCE[u] ← ‖ℓ(PATH[v] ∪ (v, u) ∪ E′)‖_p
13:        end if
14:    end for
15:    Q ← Q \ {v}
16: end while
17: return PATH[t]
```

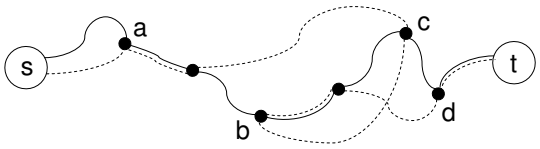

Figure 1 The definition of non-redundant merging nodes used in the proof of Lemma 13. The solid lines represent the (s,t)-path S returned by Algorithm 2, while the dashed ones represent the optimal solution S^* (solid and dashed lines are drawn adjacently when some set of edges are shared by S and S^*). We have $M(S,S^*) = \{s,a,b,c,d,t\}$ and $\overline{M}(S,S^*) = \{s,a,b,d,t\}$ since node c is redundant.

Let $(S_1,\ldots,S_k)$ and $(S_1^*,\ldots,S_k^*)$ be the k-partitions of S and $\overline{S}^*$, respectively, such that, for every $i \in [k]$, S_i (resp., S_i^*) is the set of edges connecting $\overline{v}_{i-1}$ to $\overline{v}_i$ in S (resp., S^*).

The well-known semantics of Dijkstra's algorithm guarantees that, for every $i \in [k]$, the set of edges S_i satisfies the inequality

$$\mathsf{DISTANCE}[\overline{v}_i] \leq \|\boldsymbol{\ell}(\mathsf{PATH}[\overline{v}_{i-1}] \cup S_i^*)\|_p,$$

where, by construction, $\mathsf{DISTANCE}[\overline{v}_i] = \|\boldsymbol{\ell}(S_{\leq i})\|_p$ and

$$\|\boldsymbol{\ell}(\mathsf{PATH}[\overline{v}_{i-1}] \cup S_i^*)\|_p = \|\boldsymbol{\ell}(S_{\leq i-1} \cup S_i^*)\|_p \leq \|\boldsymbol{\ell}(S_{\leq i-1}) + \boldsymbol{\ell}(S_i^*)\|_p .$$

Hence, the claim follows. ◀

By combining Lemma 13 with Theorem 10, we obtain the following result.

▶ **Theorem 14.** *Algorithm 2 approximates* SMP *within a factor of* $O(\min\{p, \log d\})$.

We conclude this section by showing a fundamental lemma that will allow us to use Algorithm 2 as a subroutine of approximation algorithms for other MMRSs defined on graphs.

▶ **Lemma 15.** *Given an instance $I \in$ MMRS, let $(S_1, \ldots, S_k)$ be a k-decomposition of a feasible solution S for I and $(S_1^*, \ldots, S_k^*)$ be a β-intersecting partial k-decomposition of an optimal solution S^* for I. If, for every $i \in [k]$, there exists an instance $I_i = (G_i, d, s_i, t_i, p) \in$ SMP such that S_i is an (s_i, t_i)-path in G_i computed by using Algorithm 2 executed with parameters G_i, d, s_i, t_i, p and $S_{\leq i-1}$, and S_i^* is an (s_i, t_i)-path in G_i, then S possesses the β-PAID property.*

Proof. The proof is a direct extension of the proof of Lemma 13. In fact, it suffices to exploit the decomposition technique used therein within the decompositions $(S_1, \ldots, S_k)$ and $(S_1^*, \ldots, S_k^*)$. Toward this end, denoting by $(S_{i,1}, \ldots, S_{i,h_i})$ and $(S_{i,1}^*, \ldots, S_{i,h_i}^*)$ the h_i-partitions of S_i and S_i^*, respectively, that are obtained as in the proof of Lemma 13 and setting $m = \sum_{\in[k]} h_i$, we have that

$$(S_{1,1}, \ldots, S_{1,h_1}, \ldots, S_{k,1}, \ldots, S_{k,h_k}) = (T_1, \ldots, T_m)$$

is an m-decomposition of S and that

$$(S_{1,1}^*, \ldots, S_{1,h_1}^*, \ldots, S_{k,1}^*, \ldots, S_{k,h_k}^*) = (T_1^*, \ldots, T_m^*)$$

is a β-intersecting partial m-decomposition of S^*. By the same arguments used in the proof of Lemma 13, we obtain that $\|\boldsymbol{\ell}(T_{\leq i})\|_p \leq \|\boldsymbol{\ell}(T_{\leq i-1} \cup T_i^*)\|_p \leq \|\boldsymbol{\ell}(T_{\leq i-1}) + \boldsymbol{\ell}(T_i^*)\|_p$ for each $i \in [m]$, thus proving the claim. ◀

6 Minimum Multidimensional Steiner Tree

Given an undirected graph $G = (V, E)$, in which every edge $e \in E$ is associated with a d-dimensional weight $\boldsymbol{c}_e \in \mathbb{R}_+^d$, a set of $r+1$ required nodes $N = \{v_1, \ldots, v_{r+1}\} \subseteq V$, and a value $p \in \overline{\mathbb{R}}$, the minimum multidimensional Steiner tree (MMST) problem is the restriction of MMRS to instances with $R = E$ and such that $\mathcal{F}$ is the set of all trees in G whose set of nodes contains N.

We propose Algorithm 3, a Kruskal-like algorithm which takes as input the graph G, the integer d, the set of required nodes N and the value p, and uses Algorithm 2 and the functions *set*, *merge* and *prune* as subroutines. Given a required node v and an h-partition $P = \{P_1, P_2, \ldots, P_h\}$ of N, function *set* returns the set $P_i \in P$ such that $v \in P_i$; given a partition $P = \{P_1, P_2, \ldots, P_h\}$ of N and two sets $P_i, P_j \in P$, function *merge* returns the partition of N obtained from P by merging P_i and P_j; finally, given a set of edges S, function *prune* returns a maximal set of edges $S' \subseteq S$ not inducing cycles in G.

The following lemma characterizes the approximation guarantee achieved by Algorithm 3.

▶ **Lemma 16.** *Fix an instance $I = (G, d, N, p) \in$ MMST. Then, Algorithm 3 returns a feasible solution for I possessing the* 2-PAID *property.*

Proof. By the well-known semantics of Kruskal's algorithm, we have that the while-loop at lines 4-12 of Algorithm 3 is executed exactly r times so that $S = (S_1, \ldots, S_r)$ induces a subgraph of G spanning N and $\overline{S} = prune(S)$ is a Steiner tree spanning N. This implies that the set of edges $\overline{S}$ returned by Algorithm 3 is a feasible solution for I. We shall prove that S possesses the 2-PAID property which, given that $\overline{S} \subset S$, implies the claim.

Let S^* be an optimal solution to I. Our proof is based on the following idea: for every $i \in [r]$, we associate a path $g(i) \in S^*$ to path $\pi(s_i, t_i) = S_i$ such that every edge in S^* appears at most twice along all sets of paths $\{g(1), \ldots, g(r)\}$. We shall prove that, using this function, we obtain an r-decomposition of S and a 2-intersecting r-decomposition of

Algorithm 3 (G, d, N, p)

1: $S_0 \leftarrow \emptyset$
2: $i \leftarrow 1$
3: $P_i \leftarrow \{\{v_1\}, \ldots, \{v_{r+1}\}\}$
4: **while** $P_i \neq N$ **do**
5: **for** each $(s,t) \in N^2 : set(s, P_i) \neq set(t, P_i)$ **do**
6: $\pi(s,t) \leftarrow$ Algorithm 2$(G, d, s, t, p, S_{\leq i-1})$
7: **end for**
8: $(s_i, t_i) \leftarrow \text{argmin}_{(s,t) \in N^2 : set(s,P_i) \neq set(t,P_i)} \{\|\boldsymbol{\ell}(S_{\leq i-1} \cup \pi(s,t))\|_p\}$
9: $S_i \leftarrow \pi(s_i, t_i)$
10: $P_{i+1} \leftarrow merge(P_i, set(P_i, s_i), set(P_i, t_i))$
11: $i \leftarrow i + 1$
12: **end while**
13: **return** $prune(S_{\leq i-1})$

S^*, meeting the conditions required in order to apply Lemma 15; the claim will then follow directly.

Towards this end, given two required nodes $u, v \in N$, let $\Sigma_{uv} = \{\pi^1_{uv}, \ldots, \pi^{p_{uv}}_{uv}\}$ denote the set of all (u,v)-paths in G. Define $\widetilde{G} = (N, \widetilde{E})$ as the multi-graph such that there are p_{uv} edges $\{\widetilde{e}^1_{uv}, \ldots, \widetilde{e}^{p_{uv}}_{uv}\}$ between every pair of nodes $u, v \in N$, with $\boldsymbol{c}_{\widetilde{e}^i_{uv}} = \boldsymbol{\ell}(\pi^i_{uv})$ for every $i \in [p_{uv}]$; so, there is a cost-preserving bijection between edges in $\widetilde{G}$ and paths in G. Given a path π in G, denote by $\widetilde{e}(\pi)$ its correspondent edge in $\widetilde{G}$ and, vice-versa, given an edge $\widetilde{e}$ of $\widetilde{G}$, denote by $\rho(\widetilde{e})$ its correspondent path in G.

We observe the following facts:

1. S induces a spanning tree $T(S)$ for $\widetilde{G}$ defined as $T(S) = \{\widetilde{e}(\pi(s_i, t_i)) \in \widetilde{E} : i \in [r]\}$.
2. S^* induces a spanning tree $T(S^*)$ for $\widetilde{G}$ defined as follows: let $(v_1, \ldots, v_{r+1})$ be the ordered sequence of the $r+1$ nodes in N listed according to the order in which appear along a Depth First Search of S^* starting from an arbitrary required node $v_1 \in N$, then $T(S^*) = \{\widetilde{e}(\pi^*_i) : i \in [r]\}$, where π^*_i is the (v_i, v_{i+1})-path in S^*. It is well-known that every edge in S^* occurs at most twice in the set of paths $\{\widetilde{e}(\pi^*_1), \ldots, \widetilde{e}(\pi^*_r)\}$.
3. Since both $T(S)$ and $T(S^*)$ are bases of the graphic matroid defined over $\widetilde{G}$, by applying Proposition 6, there exists a bijection $f : T(S) \setminus T(S^*) \to T(S^*) \setminus T(S)$.

Let us define a function $g : [r] \to S^*$ such that, for every $i \in [r]$,

$$g(i) = \begin{cases} \pi(s_i, t_i) & \text{if } \widetilde{e}(\pi(s_i, t_i)) \in T(S^*), \\ \rho(f(\widetilde{e}(\pi(s_i, t_i)))) & \text{if } \widetilde{e}(\pi(s_i, t_i)) \notin T(S^*). \end{cases}$$

For every $i \in [r]$, set $S^*_i = g(i)$. We have that $(S_1, \ldots, S_r)$ is an r-decomposition of S and $(S^*_1, \ldots, S^*_r)$ is a 2-intersecting r-decomposition of S^*. Our aim now is to apply Lemma 15.

Towards this end, fix an index $i \in [r]$ and denote by s^*_i and t^*_i the two endpoints of path $g(i)$. We observe that it must be $set(s^*_i, P_i) \neq set(t^*_i, P_i)$. Indeed, if this is not the case, then $(T(S) \setminus \widetilde{e}(\pi(s_i, t_i)) \cup \widetilde{e}(g(i))$ cannot be a basis of the graphic matroid defined over $\widetilde{G}$.

At the ith iteration of the while-loop at lines 4-12 of Algorithm 3, P_i represents the set of connected components of G induced by the set of edges $S_{\leq i-1}$. Define $\overline{G}_i$ as the multi-graph obtained from G by contracting the connected components containing $set(s_i, P_i)$ and $set(s^*_i, P_i)$ into a super-node $\overline{s}_i$ and the connected components containing $set(t_i, P_i)$ and $set(t^*_i, P_i)$ into a super node $\overline{t}_i$. Let G_i be the graph obtained from $\overline{G}_i$ by splitting every

multi-edge $e = \{u_i, u_j\}$ of weight $\boldsymbol{c}_e$ into two edges $\{u_i, u_{ij}\}$ and $\{u_{ij}, u_j\}$ of weights $\boldsymbol{c}_e$ and $\mathbf{0}^d$, respectively.

If $set(s_i, P_i) = set(s_i^*, P_i)$ and $set(t_i, P_i) = set(t_i^*, P_i)$, we have that $(G_i, \overline{s}_i, \overline{t}_i, p)$ is an instance of the SMP meeting the conditions of Lemma 15, by which, we obtain the claim. If $set(s_i, P_i) = set(s_i^*, P_i)$ and $set(t_i, P_i) = set(t_i^*, P_i)$ does not hold, since G is an undirected graph, we may assume without loss of generality that $set(s_i^*, P_i) \neq set(t_i, P_i)$ and $set(s_i, P_i) \neq set(t_i^*, P_i)$ (in fact, if one of the two inequalities does not hold, we can exchange the role of s_i^* and t_i^* and have both of them satisfied).

Now observe that there is a cost-preserving bijection b between the set of paths connecting any node in $set(s_i, P_i) \cup set(s_i^*, P_i)$ to any node in $set(t_i, P_i) \cup set(t_i^*, P_i)$ in G and the set of $(\overline{s}_i, \overline{t}_i)$-paths in G_i. Moreover, by the assumption $set(s_i, P_i) \neq set(t_i^*, P_i)$, it follows that path $b(S_i)$ is an $(\overline{s}_i, \overline{t}_i)$-path in G_i; similarly, by the assumption $set(s_i^*, P_i) \neq set(t_i, P_i)$, it follows that path $b(g_i)$ is an $(\overline{s}_i, \overline{t}_i)$-path in G_i.

Thus, in order to apply Lemma 15 and obtain the claim, we need to prove that there are suitable tie breaking rules for which $b(S_i)$ is the output of Algorithm 2 when executed with parameters G_i, d, $\overline{s}_i$, $\overline{t}_i$, p and $S_{\leq i-1}$ on the instance $(G_i, d, \overline{s}_i, \overline{t}_i, p) \in$ SMP. Assume, by way of contradiction, that for any possible tie breaking rule, Algorithm 2 never returns an $(\overline{s}_i, \overline{t}_i)$-path π_i such that $\pi_i = b(S_i)$. This implies that there exists an $(\overline{s}_i, \overline{t}_i)$-path π_i^* such that $\|\boldsymbol{\ell}(S_{\leq i-1} \cup \pi_i^*)\|_p < \|\boldsymbol{\ell}(S_{\leq i-1} \cup \pi_i)\|_p$ by which we obtain $\|\boldsymbol{\ell}(S_{\leq i-1} \cup b^{-1}(\pi_i^*))\|_p < \|\boldsymbol{\ell}(S_{\leq i-1} \cup b^{-1}(\pi_i))\|_p$, where $b^{-1}(\pi_i^*)$ is some (s,t)-path in G with $set(s, P_i) \neq set(t, P_i)$. This contradicts $S_i = \text{argmin}_{(s,t) \in N^2 : set(s,P_i) \neq set(t,P_i)} \{\|\boldsymbol{\ell}(S_{\leq i-1} \cup \pi(s,t))\|_p\}$. ◀

By combining Lemma 16 with Theorem 10, we obtain the following result.

▶ **Theorem 17.** *Algorithm 3 approximates* MMST *within a factor of* $O(\min\{p, \log d\})$.

7 Minimum Multidimensional Arborescence

Given a directed connected graph $G = (V, E)$, with $|V| = n$, in which every edge $e \in E$ is associated with a d-dimensional weight $\boldsymbol{c}_e \in \mathbb{R}_+^d$, a node $s \in V$ and a value $p \in \overline{\mathbb{R}}$, the minimum multidimensional arborescence (MMA) problem is the restriction of MMRS to instances with $R = E$ and such that $\mathcal{F}$ is the set of all the directed trees in G rooted at s. Recall that a directed tree $T \subset E$ rooted at s is a set of $n-1$ edges such that, for every $t \in V$, there exists a directed (s,t)-path in T.

A *rooted weakly connected component* of G is a subgraph $H = (V', E')$ of G possessing at least one root, that is, a node from which it is possible to reach any other node in V' by following a direct path in E'. Call the *representative node* of a rooted weakly connected component any of its roots. When the set of roots of a rooted weakly connected component H contains s, the representative node of H is always assumed to be s.

We propose Algorithm 4 which, starting from the set of rooted weakly connected components of G obtained by considering all nodes in V as singletons, repeatedly merges rooted weakly connected components of G until an arborescence rooted at s is obtained. Given a set of edges S, function *repr* first computes the set C of weakly connected components of G induced by S, and then computes a representative for every element of C. Observe that, by our assumption, $s \in repr(C)$ for every set of rooted weakly connected components C. Given a set of edges S and a node u, function *nodes* computes the set of nodes belonging to the weakly connected component containing u. Finally, given a set of edges S, function *prune* returns a maximal subset of S not inducing cycles in G.

The following lemma characterizes the approximation guarantee achieved by Algorithm 4. Its proof is omitted due to lack of space.

Algorithm 4 (G, d, s, p)

$S_0 \leftarrow \emptyset$
$C_1 \leftarrow V$
$i \leftarrow 1$
while $C_i \neq \{s\}$ **do**
  $S_i \leftarrow S_{i-1}$
  for each $u \in C_i \setminus \{s\}$ **do**
    for each $v \in V \setminus nodes(u, S_{i-1})$ **do**
      $\pi(v, u) \leftarrow Algorithm\ 2(G, d, v, u, p, S_i)$
    end for
    $\delta(u) \leftarrow \text{argmin}_{v \in V \setminus nodes(u, S_{i-1})}\{\|\boldsymbol{\ell}(S_i \cup \pi(v, u))\|_p\}$
    $S_i \leftarrow S_i \cup \pi(\delta(u), u)$
  end for
  $C_{i+1} \leftarrow repr(S_i)$
  $i \leftarrow i + 1$
end while
return $prune(S_{i-1})$

▶ **Lemma 18.** *Fix an instance $I = (G, d, s, p) \in$ MMA. Then, Algorithm 4 returns a feasible solution for I possessing the $O(\log n)$-PAID property.*

By combining Lemma 18 with Theorem 10, we obtain the following result.

▶ **Theorem 19.** *Algorithm 4 approximates* MMA *within a factor of $O(\log n \cdot \min\{p, \log d\})$.*

8 An Inapproximability Result

We conclude by complementing our positive algorithmic results with the following hardness statement.

▶ **Theorem 20.** *For every constant $\kappa \geq 1$, the problems* MMRS(∞) *on matroids,* SMP(∞), MMST(∞), *and* MMA(∞) *cannot be approximated up to a factor κ unless* NP=ZPP.

Proof. Our proof defines a simple approximation-preserving reduction from the Vector Scheduling Problem VSP. An instance of the VSP is defined by n tasks to be scheduled on m identical machines. Every task i has a d-dimensional load vector $\boldsymbol{c}_i \in \mathbb{R}^d_+$ and the objective is to minimize the load over all machines and all dimensions. Chekuri and Kanna [7] showed that, for every constant $\kappa \in \mathbb{R}$, this problem cannot be approximated up to a factor κ unless NP=ZPP.

Given an instance of the VSP, we define an undirected multi-graph $G = (V, E)$ with $n+1$ nodes and mn edges defined as follows: $V = \{v_0, v_1, \ldots, v_n\}$, and for every $i \in [n-1]$ there are m edges $e_i^1, \ldots, e_i^m \in E$ connecting nodes v_{i-1} and v_i such that, for every $j \in [m]$, edge e_i^j has a (dm)-dimensional cost

$$\boldsymbol{c}(e_i^j) = (\underbrace{\mathbf{0}^d, \ldots, \mathbf{0}^d}_{j-1 \text{ times}}, \boldsymbol{c}_i, \underbrace{\mathbf{0}^d, \ldots, \mathbf{0}^d}_{m-j \text{ times}}).$$

Observe that every schedule of the n tasks to the m machines corresponds to a (v_0, v_n)-path in G and viceversa. Moreover, the objective value of the two solutions is exactly the same. Since G can be easily translated into a graph G' by splitting every multi-edge into two edges of the same total cost, we have that the hardness result for the VSP extends also

to $\mathsf{SMP}(\infty)$. Given that every spanning tree for G' is a (v_0, v_n)-path in G', the hardness result extends to $\mathsf{MMRS}(\infty)$ on matroids and to $\mathsf{MMST}(\infty)$ when the set of required nodes contains both v_0 and v_n. Finally, by directing every edge in E from v_{i-1} to v_i, we obtain the same hardness result for $\mathsf{MMA}(\infty)$. ◀

References

1 https://www.participatorybudgeting.org/

2 B. Awerbuch, Y. Azar, E. F. Grove, M.-Y. Kao, P. Krishnan, and J. S. Vitter. Load balancing in the L_p norm. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 383–391, 1995.

3 A. Berger, V. Bonifaci, F. Grandoni, G. Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 128(1-2): 355–372, 2011.

4 V. Bilò, V. Goyal, R. Ravi, and M. Singh. On the crossing spanning tree problem. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Problems (APPROX)*, LNCS 3122, pages 51–60, 2004.

5 I. Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 972–982, 2008.

6 I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. *Algorithmica*, 66(3): 512–540, 2013.

7 C. Chekuri and S. Khanna. On multidimensional packing problems. *SIAM Journal on Computing*, 33(4): 837–851, 2004.

8 C. Chekuri, J. Vondrák and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 575–584, 2010.

9 C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding for matroid polytopes and applications. *arXiv*: 0909.4348, 2009.

10 C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1080–1097, 2011.

11 I. Diakonikolas, M. Yannakakis. Small approximate Pareto sets for biobjective shortest paths and other problems. *SIAM Journal on Computing*, 39(4): 1340-1371, 2009.

12 F. Grandoni, R. Ravi, M. Singh, and R. Zanklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1): 525–554, 2014.

13 R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operation Research*, 17(1): 36–42, 1992.

14 R. Hassin, Asaf Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2): 261–268, 2004.

15 D. Lorenz, D. Raz. A simple efficient approximation scheme for the restricted shortest paths problem. *Operations Research Letters*, 28: 213–219, 2001.

16 C. H. Papadimitriou, M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 86–92, 2000.

17 R. Ravi, M. Goemans. The constrained minimum spanning tree problem. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 66–75, 1996.

18 A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency. Volume B, Matroids, Trees, Stable Sets*. Springer, 2003.

19 V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.